Universidad Nacional del Sur
Departamento de Ciencias e Ingeniería de la Computación

Tecnología de Programación

(2019)

Práctico 1
Repaso

Temas: Paradigma Orientado a Objetos. Programación Orientada a Objetos. Herencia. Polimorfismo. Vinculación dinámica de código. Genericidad. Java.

Ejercicio 1. Considerando las implementaciones y la interfaz de la clase Lista vista en ED. Describa donde interviene el polimorfismo y defina si es una implementación que utiliza genericidad o no y por qué.

Ejercicio 2. ¿Cuál es el rol de una interfaz en el Paradigma Orientado a Objetos? ¿Cuál es la diferencia entre una interfaz y una clase abstracta?

Ejercicio 3. Implementar en Java, utilizando genericidad, la clase ParOrdenado[A,B]. Es decir, los tipos A y B son parámetros formales de la clase ParOrdenado.

Ejercicio 4. Implementar en Java, utilizando genericidad, la clase Zipper[A,B], donde A y B son parámetros formales de la clase. La clase debe proveer un único método de instancia llamado Zip con la signatura:

Zip(Xs: Secuencia[A]; Ys: Secuencia[B]) : Secuencia[ParOrdenado[A,B]].

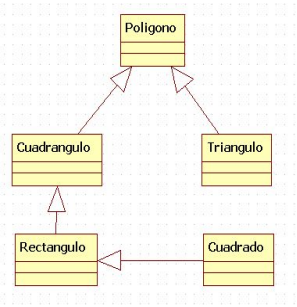
El método Zip recibe dos secuencias Xs e Ys y retorna la secuencia de los pares de elementos formados por los elementos de Xs e Ys de acuerdo a la siguiente especificación:

Zip(<a1, a2, a3, ….>, <b1, b2, b3, …>) --> <(a1,b1), (a2,b2), (a3,b3),….>

En el caso en que las listas tengan distinta longitud, la longitud de la lista resultante es la longitud de la lista más corta.

En el caso de las Listas utilice las interfaces e implementaciones que vienen por defecto en Java, por ejemplo utilizando “ArrayList”.

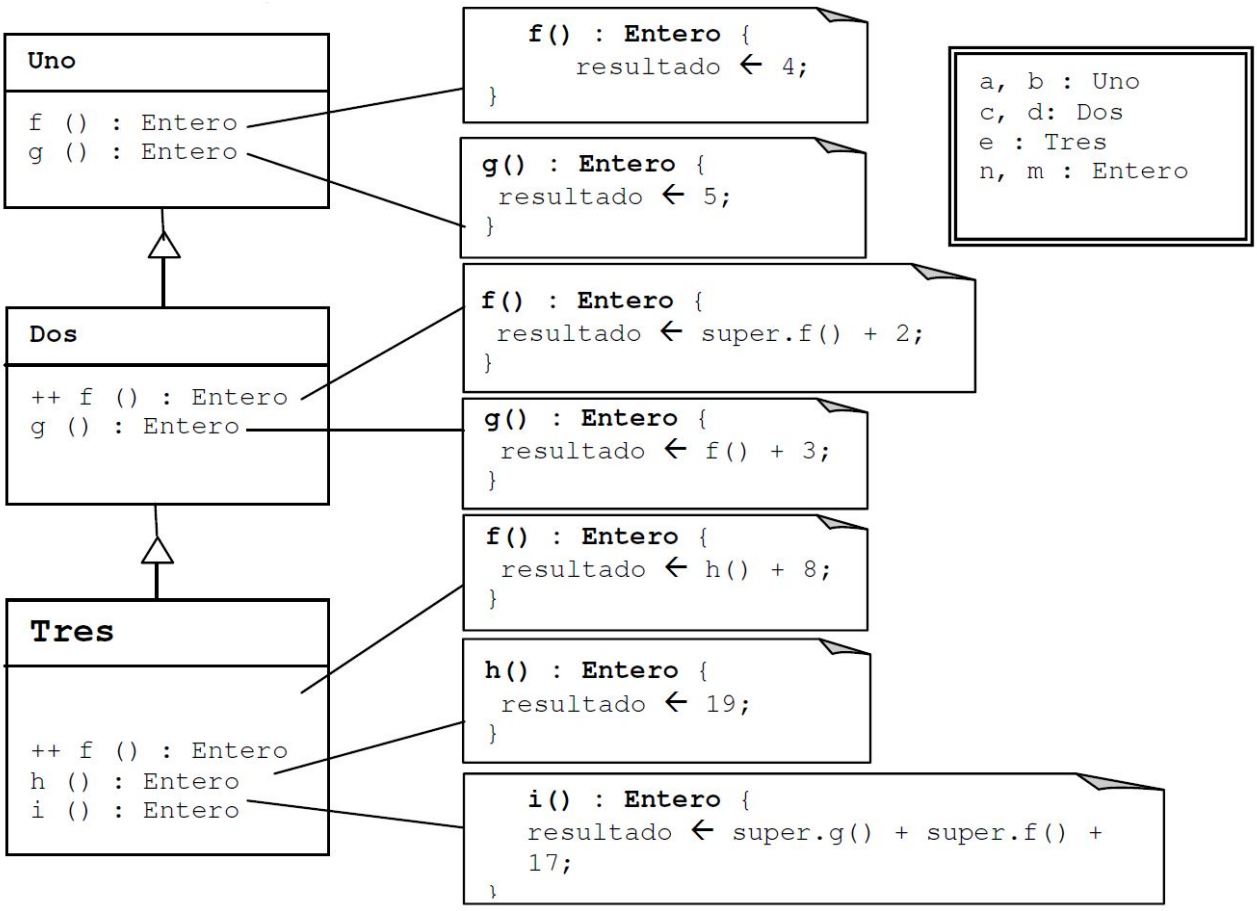
Ejercicio 5. Teniendo en cuenta la siguiente jerarquía de clases y las siguientes declaraciones, indique cuáles de las sentencias (a,.., g) no son válidas y por qué:



t : Triangulo;
p : Poligono:
c: Cuadrangulo;
r: Rectangulo;
cd: Cuadrado;

(a) p := c;
(b) p := r;
(c) cd := c;
(d) c := t;
(e) {p := r; r := p;}
(f) p.diagonal(); //diagonal es una propiedad de los cuadrángulos
(g) t.perimetro(); //perímetro es una propiedad de los polígonos

Ejercicio 6. Considerando las siguientes definiciones de clases y de referencias; determinar el valor de n y m en todos los casos donde sea posible (Programa 1, 2 y 3). Indicar si existen errores de compilación o ejecución. En caso de existir errores, ejecute hasta la última sentencia válida. Siempre justificar apropiadamente cada respuesta.



Programa 1	Programa 2	Programa 3
(1) a := !!Uno(); (2) b := a; (3) d := !!Dos(); (4) a := d; (5) m := a.g(); (6) c := b; (7) n := c.g();	(1) a := !!Uno(); (2) c := !!Dos(); (3) a := c; (4) n := a.f(); (5) m := n + a.h();	(1) e := !!Tres(); (2) b := !!Dos(); (3) c := !!Dos(); (4) a := e; (5) n := a.f() + e.i(); (6) e := (Tres)b; (7) m := n + e.h();